FAX

ATTN. Michael D. Meucci

Fax Number 1 571 273 8300

Phone Number 571 272 3892

FROM Volel Emile, Esq.

Fax Number 512 306 0240

Phone Number 512 306 7969

SUBJECT Response to Non-Compliant Appeal Brief

Number of Pages 53

Date 3/31/2006

MESSAGE

This fax communication contains:

1. one copy of a Fax Transmittal Form; and
3. three copies of the Response to Notice of Non-Compliant Appeal Brief.

Volel

MAR 31 2006

PTO/SB/21 (02-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	09/965,002
Filing Date	09/27/2001
First Named Inventor	Sanaa F. Abdelhadi
Art Unit	2142
Examiner Name	Michael D. Mrucci
Attorney Docket Number	AUS920010905US1

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation
- ☐ Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
- ☐ After Allowance communication to Technology Center (TC)
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please identify below):

Remarks

Response to Notice of Non-Compliant Appeal Brief

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Volel Emile
Signature	
Date	03/31/2006

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	Volel Emile		
Signature		Date	03/31/2006

This collection of information is required by 37 CFR 1.6. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Application of: Abdelhadi et. al.	:	
	:	Before the Examiner:
Serial No: 09/965,002	:	Michael D. Meucci
	:	
Filed: 09/27/2001	:	Group Art Unit: 2142
	:	
Title: APPARATUS AND METHOD OF REPRESENTING REAL-TIME DISTRIBUTED COMMAND EXECUTION STATUS ACROSS DISTRIBUTED SYSTEMS	:	Confirmation No.: 2728

RESPONSE TO NOTICE OF NON-COMPLAINT APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is a Response to a Notice of Non-Compliant Appeal Brief dated March 16, 2006.

BRIEF FOR APPLICANTS - APPELLANTS

(i)

Real Party in Interest

The real party in interest is International Business Machines Corporation (IBM), the assignee.

(ii)

Related Appeals and Interferences

There are no other appeals or interferences known to appellants, appellants' representative or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii)

Status of Claims

Claims 1 - 37 have been finally rejected under 35 U.S.C. §103 as being unpatentable over Joyce et al. in view of Ahmed et al. in an Office Action dated August 12, 2005.

(iv)

Status of Amendment

All amendments have been entered.

(v)

Summary of Claimed Subject Matter

In accordance with the teachings of the invention, when a command is being executed on a plurality of computer systems on a network, a dialog window is displayed (page 17, lines 9 - 16 and Fig. 10). In the dialog window, sub-

windows for displaying present status of the execution of the command on each of the computer systems are displayed (page 17, lines 24 to page 18, line 9).

(vi)

Grounds of Rejection to be Reviewed on Appeal

Whether Claims 1 - 37 were properly rejected under 35 U.S.C. §103(a) as being unpatentable over Joyce et al. in view of Ahmed et al.

(vii)

Arguments

In considering a Section §103 rejection, the subject matter of the claim "as a whole" must be considered and analyzed. In the analysis, it is necessary that the scope and contents of the prior art and differences between the art and the claimed invention be determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966).

Joyce et al. purport to teach a method of using monitoring tools to support the development of distributed systems that interact via message passing (see page 122, lines 1 - 3). In accordance with the teachings of Joyce et al., Jade, a programming environment, is used to support the development of a distributed program. Jade includes a window system, a graphics package, an interactive graphics editor and a distributed monitoring system (see Section 2 on page 125).

The graphics package provides routines for creating and manipulating pictures and the graphics editor

facilitates the creation of pictures that can be used to represent specific states of an executing distributed program (see the 4th full paragraph of Section 2.1 on page 125). The window system may be used by a user to create and manipulate windows using a mouse, for example. A window is a virtual terminal as well as an interface to Jade processes (see the 3rd full paragraph of Section 2.1 on page 125).

Thus, in conjunction with the window system, the graphics package and the graphics editor, the distributed monitoring system may be used to observe a set of Jade processes executing on different machines (see the 1st full paragraph of Section 2.1 on page 125 as well as the 1st full paragraph of Section 2.2 on page 126).

The system may be set such that each time an event is received (which is generally done through message passing from one computer system to another), a picture that represents a current state of the inter-process communication of the distributed application program is updated and displayed to the user (see Section 3.2 on pages 133 and 134). Consequently, an animated graphical view of an event stream, such as that shown in Fig. 7, may be displayed to a user.

But Joyce et al. do not teach, show or suggest the step of ***displaying a dialog window that is divided into sub-windows in which the status of a command that is being executed on a plurality of computer systems is displayed*** as claimed.

Ahmed et al. purport to teach a distributed framework for intertask communication between workstation

applications. According to the purported teachings of Ahmed et al., one or more workstations are interconnected by an extensible intertask communication (ITC) apparatus. Each workstation has a display in which one or more windows are presented to an operator. Each window is generated in response to the execution of an application program or client application. Each client application has a Human Interface Code and a Framework Code. The Framework Code, in conjunction with a server program, transmits and communicates event information directly between a first client application and a second client application, or a plurality of client application programs concurrently executing in one or more workstations of a network of interconnected workstations, without requiring that event information pass through and register with an intervening server or dispatcher application program, if and when an interest object is initially transmitted between the first client application and the second client application via the server program.

An event is an action taken by one operator at a workstation. For example, that operator may drag the cursor by moving a mouse or perhaps the operator will delete data or create new data. That event information, being practiced by one operator in one program application at one workstation, may be needed by another operator in another program application at another workstation. The interprocess communication can transmit that event information from the one program application to all other program applications in the network of workstations, without requiring that the event information register with

an intervening server or dispatcher program, provided that an interest object(s) was initially transmitted between the one program application and all the other program applications via a server which are concurrently executing in all of the workstations in the network of workstations.

However, just as in the case of Joyce et al., Ahmed et al. do not show, teach or so much as suggest the step of ***displaying a dialog window that is divided into sub-windows in which the status of a command that is being executed on a plurality of computer systems is displayed*** as claimed.

Since the references, neither alone nor in combination, teach, show or suggest the claimed invention, Applicants submit that the claims in the Application are allowable. Hence, Applicants respectfully request allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

(viii)

Claims Appendix

1. (Previously presented) A method of displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said method comprising the steps of:

displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

displaying the status of the execution of the command on each of the computer systems within a proper sub-window.

2. (Original) The method of Claim 1 wherein said sub-windows include a "waiting" sub-window, a "working" sub-window and a "completed" sub-window.

3. (Original) The method of Claim 2 wherein the step of displaying the status of the execution of the command includes displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

4. (Original) The method of Claim 3 wherein when the command begins to execute on a computer system, the

name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

5. (Original) The method of Claim 4 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

6. (Original) The method of Claim 5 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

7. (Original) The method of Claim 6 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

8. (Previously presented) The method of Claim 7 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

9. (Previously presented) The method of Claim 8 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

10. (Original) The method of Claim 9 wherein when the displayed name of a computer system is selected

further information about the status of the command executing on the computer system is displayed.

11. (Original) The method of Claim 10 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

12. (Previously presented) The method of Claim 11 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

13. (Previously presented) A computer program product on a computer readable medium for displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said computer program product comprising:

code for displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

code for displaying the status of the execution of the command on each of the computer systems within the proper sub-window.

14. (Original) The computer program product of Claim 13 wherein said sub-windows include a "waiting" sub-

window, a "working" sub-window and a "completed" sub-window.

15. (Original) The computer program product of Claim 14 wherein the code for displaying the status of the execution of the command includes code for displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

16. (Original) The computer program product of Claim 15 wherein when the command begins to execute on a computer system, the name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

17. (Original) The computer program product of Claim 16 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

18. (Original) The computer program product of Claim 17 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

19. (Original) The computer program product of Claim 18 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

20. (Previously presented) The computer program product of Claim 19 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

21. (Previously presented) The computer program product of Claim 20 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

22. (Original) The computer program product of Claim 21 wherein when the displayed name of a computer system is selected further information about the status of the command executing on the computer system is displayed.

23. (Original) The computer program product of Claim 22 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

24. (Previously presented) The computer program product of Claim 23 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

25. (Previously presented) An apparatus for displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said apparatus comprising:

means for displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

means for displaying the status of the execution of the command on each of the computer systems within the proper sub-window.

26. (Original) The apparatus of Claim 25 wherein said sub-windows include a "waiting" sub-window, a "working" sub-window and a "completed" sub-window.

27. (Original) The apparatus of Claim 26 wherein the means for displaying the status of the execution of the command includes means for displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

28. (Original) The apparatus of Claim 27 wherein when the command begins to execute on a computer system, the name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

29. (Original) The apparatus of Claim 28 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

30. (Original) The apparatus of Claim 29 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

31. (Original) The apparatus of Claim 30 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

32. (Previously presented) The apparatus of Claim 31 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

33. (Previously presented) The apparatus of Claim 32 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

34. (Original) The apparatus of Claim 33 wherein when the displayed name of a computer system is selected further information about the status of the command executing on the computer system is displayed.

35. (Original) The apparatus of Claim 34 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

36. (Previously presented) The apparatus of Claim 35 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

37. (Previously presented) A method of displaying an execution status of a command, the command being executed by a plurality of computer systems on a network, the computer systems running different system management software utilities having different command structures, the method comprising the steps of:

enabling a user to enter the command in a common interface, the command being either a request to start execution of another command or to stop execution of the other command, the common interface translating the command into the different command structures;

enabling a user to send the command to the plurality of the computer systems;

enabling a user to indicate whether or not the execution of the command is to be monitored;

displaying, if the execution of the command is to be monitored, a dialog window that is divided into a waiting, working, successful and failed sub-windows for displaying present status of the execution of the command on each of the computer systems executing the command; and

displaying the status of the execution of the command on each of the computer systems within a proper sub-window.

(IX)

Evidence Appendix

None.

(x)

Related Proceedings Appendix

None.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Application of: Abdelhadi et. al.	:	
	:	Before the Examiner:
Serial No: 09/965,002	:	Michael D. Meucci
	:	
Filed: 09/27/2001	:	Group Art Unit: 2142
	:	
Title: APPARATUS AND METHOD OF REPRESENTING REAL-TIME DISTRIBUTED COMMAND EXECUTION STATUS ACROSS DISTRIBUTED SYSTEMS	: Confirmation No.: 2728	

RESPONSE TO NOTICE OF NON-COMPLAINT APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is a Response to a Notice of Non-Compliant Appeal Brief dated March 16, 2006.

BRIEF FOR APPLICANTS - APPELLANTS

(i)

Real Party in Interest

The real party in interest is International Business Machines Corporation (IBM), the assignee.

(ii)

Related Appeals and Interferences

There are no other appeals or interferences known to appellants, appellants' representative or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii)

Status of Claims

Claims 1 - 37 have been finally rejected under 35 U.S.C. §103 as being unpatentable over Joyce et al. in view of Ahmed et al. in an Office Action dated August 12, 2005.

(iv)

Status of Amendment

All amendments have been entered.

(v)

Summary of Claimed Subject Matter

In accordance with the teachings of the invention, when a command is being executed on a plurality of computer systems on a network, a dialog window is displayed (page 17, lines 9 - 16 and Fig. 10). In the dialog window, sub-

windows for displaying present status of the execution of the command on each of the computer systems are displayed (page 17, lines 24 to page 18, line 9).

(vi)

Grounds of Rejection to be Reviewed on Appeal

Whether Claims 1 - 37 were properly rejected under 35 U.S.C. §103(a) as being unpatentable over Joyce et al. in view of Ahmed et al.

(vii)

Arguments

In considering a Section §103 rejection, the subject matter of the claim "as a whole" must be considered and analyzed. In the analysis, it is necessary that the scope and contents of the prior art and differences between the art and the claimed invention be determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966).

Joyce et al. purport to teach a method of using monitoring tools to support the development of distributed systems that interact via message passing (see page 122, lines 1 - 3). In accordance with the teachings of Joyce et al., Jade, a programming environment, is used to support the development of a distributed program. Jade includes a window system, a graphics package, an interactive graphics editor and a distributed monitoring system (see Section 2 on page 125).

The graphics package provides routines for creating and manipulating pictures and the graphics editor

facilitates the creation of pictures that can be used to represent specific states of an executing distributed program (see the 4th full paragraph of Section 2.1 on page 125). The window system may be used by a user to create and manipulate windows using a mouse, for example. A window is a virtual terminal as well as an interface to Jade processes (see the 3rd full paragraph of Section 2.1 on page 125).

Thus, in conjunction with the window system, the graphics package and the graphics editor, the distributed monitoring system may be used to observe a set of Jade processes executing on different machines (see the 1st full paragraph of Section 2.1 on page 125 as well as the 1st full paragraph of Section 2.2 on page 126).

The system may be set such that each time an event is received (which is generally done through message passing from one computer system to another), a picture that represents a current state of the inter-process communication of the distributed application program is updated and displayed to the user (see Section 3.2 on pages 133 and 134). Consequently, an animated graphical view of an event stream, such as that shown in Fig. 7, may be displayed to a user.

But Joyce et al. do not teach, show or suggest the step of ***displaying a dialog window that is divided into sub-windows in which the status of a command that is being executed on a plurality of computer systems is displayed*** as claimed.

Ahmed et al. purport to teach a distributed framework for intertask communication between workstation

applications. According to the purported teachings of Ahmed et al., one or more workstations are interconnected by an extensible intertask communication (ITC) apparatus. Each workstation has a display in which one or more windows are presented to an operator. Each window is generated in response to the execution of an application program or client application. Each client application has a Human Interface Code and a Framework Code. The Framework Code, in conjunction with a server program, transmits and communicates event information directly between a first client application and a second client application, or a plurality of client application programs concurrently executing in one or more workstations of a network of interconnected workstations, without requiring that event information pass through and register with an intervening server or dispatcher application program, if and when an interest object is initially transmitted between the first client application and the second client application via the server program.

An event is an action taken by one operator at a workstation. For example, that operator may drag the cursor by moving a mouse or perhaps the operator will delete data or create new data. That event information, being practiced by one operator in one program application at one workstation, may be needed by another operator in another program application at another workstation. The interprocess communication can transmit that event information from the one program application to all other program applications in the network of workstations, without requiring that the event information register with

an intervening server or dispatcher program, provided that an interest object(s) was initially transmitted between the one program application and all the other program applications via a server which are concurrently executing in all of the workstations in the network of workstations.

However, just as in the case of Joyce et al., Ahmed et al. do not show, teach or so much as suggest the step of ***displaying a dialog window that is divided into sub-windows in which the status of a command that is being executed on a plurality of computer systems is displayed*** as claimed.

Since the references, neither alone nor in combination, teach, show or suggest the claimed invention, Applicants submit that the claims in the Application are allowable. Hence, Applicants respectfully request allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

(viii)

Claims Appendix

1. (Previously presented) A method of displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said method comprising the steps of:

displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

displaying the status of the execution of the command on each of the computer systems within a proper sub-window.

2. (Original) The method of Claim 1 wherein said sub-windows include a "waiting" sub-window, a "working" sub-window and a "completed" sub-window.

3. (Original) The method of Claim 2 wherein the step of displaying the status of the execution of the command includes displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

4. (Original) The method of Claim 3 wherein when the command begins to execute on a computer system, the

name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

5. (Original) The method of Claim 4 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

6. (Original) The method of Claim 5 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

7. (Original) The method of Claim 6 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

8. (Previously presented) The method of Claim 7 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

9. (Previously presented) The method of Claim 8 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

10. (Original) The method of Claim 9 wherein when the displayed name of a computer system is selected

further information about the status of the command executing on the computer system is displayed.

11. (Original) The method of Claim 10 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

12. (Previously presented) The method of Claim 11 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

13. (Previously presented) A computer program product on a computer readable medium for displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said computer program product comprising:

code for displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

code for displaying the status of the execution of the command on each of the computer systems within the proper sub-window.

14. (Original) The computer program product of Claim 13 wherein said sub-windows include a "waiting" sub-

window, a "working" sub-window and a "completed" sub-window.

15. (Original) The computer program product of Claim 14 wherein the code for displaying the status of the execution of the command includes code for displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

16. (Original) The computer program product of Claim 15 wherein when the command begins to execute on a computer system, the name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

17. (Original) The computer program product of Claim 16 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

18. (Original) The computer program product of Claim 17 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

19. (Original) The computer program product of Claim 18 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

20. (Previously presented) The computer program product of Claim 19 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

21. (Previously presented) The computer program product of Claim 20 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

22. (Original) The computer program product of Claim 21 wherein when the displayed name of a computer system is selected further information about the status of the command executing on the computer system is displayed.

23. (Original) The computer program product of Claim 22 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

24. (Previously presented) The computer program product of Claim 23 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

25. (Previously presented) An apparatus for displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said apparatus comprising:

means for displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

means for displaying the status of the execution of the command on each of the computer systems within the proper sub-window.

26. (Original) The apparatus of Claim 25 wherein said sub-windows include a "waiting" sub-window, a "working" sub-window and a "completed" sub-window.

27. (Original) The apparatus of Claim 26 wherein the means for displaying the status of the execution of the command includes means for displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

28. (Original) The apparatus of Claim 27 wherein when the command begins to execute on a computer system, the name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

29. (Original) The apparatus of Claim 28 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

30. (Original) The apparatus of Claim 29 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

31. (Original) The apparatus of Claim 30 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

32. (Previously presented) The apparatus of Claim 31 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

33. (Previously presented) The apparatus of Claim 32 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

34. (Original) The apparatus of Claim 33 wherein when the displayed name of a computer system is selected further information about the status of the command executing on the computer system is displayed.

35. (Original) The apparatus of Claim 34 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

36. (Previously presented) The apparatus of Claim 35 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

37. (Previously presented) A method of displaying an execution status of a command, the command being executed by a plurality of computer systems on a network, the computer systems running different system management software utilities having different command structures, the method comprising the steps of:

enabling a user to enter the command in a common interface, the command being either a request to start execution of another command or to stop execution of the other command, the common interface translating the command into the different command structures;

enabling a user to send the command to the plurality of the computer systems;

enabling a user to indicate whether or not the execution of the command is to be monitored;

displaying, if the execution of the command is to be monitored, a dialog window that is divided into a waiting, working, successful and failed sub-windows for displaying present status of the execution of the command on each of the computer systems executing the command; and

displaying the status of the execution of the command on each of the computer systems within a proper sub-window.

(IX)

Evidence Appendix

None.

(x)

Related Proceedings Appendix

None.

RECEIVED
CENTRAL FAX CENTER

MAR 31 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Application of: Abdelhadi et. al.	:	
	:	Before the Examiner:
Serial No: 09/965,002	:	Michael D. Meucci
	:	
Filed: 09/27/2001	:	Group Art Unit: 2142
	:	
Title: APPARATUS AND METHOD OF REPRESENTING REAL-TIME DISTRIBUTED COMMAND EXECUTION STATUS ACROSS DISTRIBUTED SYSTEMS	:	Confirmation No.: 2728

RESPONSE TO NOTICE OF NON-COMPLAINT APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is a Response to a Notice of Non-Compliant Appeal Brief dated March 16, 2006.

BRIEF FOR APPLICANTS - APPELLANTS

(i)

Real Party in Interest

The real party in interest is International Business Machines Corporation (IBM), the assignee.

(ii)

Related Appeals and Interferences

There are no other appeals or interferences known to appellants, appellants' representative or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii)

Status of Claims

Claims 1 - 37 have been finally rejected under 35 U.S.C. §103 as being unpatentable over Joyce et al. in view of Ahmed et al. in an Office Action dated August 12, 2005.

(iv)

Status of Amendment

All amendments have been entered.

(v)

Summary of Claimed Subject Matter

In accordance with the teachings of the invention, when a command is being executed on a plurality of computer systems on a network, a dialog window is displayed (page 17, lines 9 - 16 and Fig. 10). In the dialog window, sub-

windows for displaying present status of the execution of the command on each of the computer systems are displayed (page 17, lines 24 to page 18, line 9).

(vi)

Grounds of Rejection to be Reviewed on Appeal

Whether Claims 1 - 37 were properly rejected under 35 U.S.C. §103(a) as being unpatentable over Joyce et al. in view of Ahmed et al.

(vii)

Arguments

In considering a Section §103 rejection, the subject matter of the claim "as a whole" must be considered and analyzed. In the analysis, it is necessary that the scope and contents of the prior art and differences between the art and the claimed invention be determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966).

Joyce et al. purport to teach a method of using monitoring tools to support the development of distributed systems that interact via message passing (see page 122, lines 1 - 3). In accordance with the teachings of Joyce et al., Jade, a programming environment, is used to support the development of a distributed program. Jade includes a window system, a graphics package, an interactive graphics editor and a distributed monitoring system (see Section 2 on page 125).

The graphics package provides routines for creating and manipulating pictures and the graphics editor

facilitates the creation of pictures that can be used to represent specific states of an executing distributed program (see the 4th full paragraph of Section 2.1 on page 125). The window system may be used by a user to create and manipulate windows using a mouse, for example. A window is a virtual terminal as well as an interface to Jade processes (see the 3rd full paragraph of Section 2.1 on page 125).

Thus, in conjunction with the window system, the graphics package and the graphics editor, the distributed monitoring system may be used to observe a set of Jade processes executing on different machines (see the 1st full paragraph of Section 2.1 on page 125 as well as the 1st full paragraph of Section 2.2 on page 126).

The system may be set such that each time an event is received (which is generally done through message passing from one computer system to another), a picture that represents a current state of the inter-process communication of the distributed application program is updated and displayed to the user (see Section 3.2 on pages 133 and 134). Consequently, an animated graphical view of an event stream, such as that shown in Fig. 7, may be displayed to a user.

But Joyce et al. do not teach, show or suggest the step of ***displaying a dialog window that is divided into sub-windows in which the status of a command that is being executed on a plurality of computer systems is displayed*** as claimed.

Ahmed et al. purport to teach a distributed framework for intertask communication between workstation

applications. According to the supported teachings of Ahmed et al., one or more workstations are interconnected by an extensible intertask communication (ITC) apparatus. Each workstation has a display in which one or more windows are presented to an operator. Each window is generated in response to the execution of an application program or client application. Each client application has a Human Interface Code and a Framework Code. The Framework Code, in conjunction with a server program, transmits and communicates event information directly between a first client application and a second client application, or a plurality of client application programs concurrently executing in one or more workstations of a network of interconnected workstations, without requiring that event information pass through and register with an intervening server or dispatcher application program, if and when an interest object is initially transmitted between the first client application and the second client application via the server program.

An event is an action taken by one operator at a workstation. For example, that operator may drag the cursor by moving a mouse or perhaps the operator will delete data or create new data. That event information, being practiced by one operator in one program application at one workstation, may be needed by another operator in another program application at another workstation. The interprocess communication can transmit that event information from the one program application to all other program applications in the network of workstations, without requiring that the event information register with

an intervening server or dispatcher program, provided that an interest object(s) was initially transmitted between the one program application and all the other program applications via a server which are concurrently executing in all of the workstations in the network of workstations.

However, just as in the case of Joyce et al., Ahmed et al. do not show, teach or so much as suggest the step of ***displaying a dialog window that is divided into sub-windows in which the status of a command that is being executed on a plurality of computer systems is displayed*** as claimed.

Since the references, neither alone nor in combination, teach, show or suggest the claimed invention, Applicants submit that the claims in the Application are allowable. Hence, Applicants respectfully request allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

(viii)

Claims Appendix

1. (Previously presented) A method of displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said method comprising the steps of:

displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

displaying the status of the execution of the command on each of the computer systems within a proper sub-window.

2. (Original) The method of Claim 1 wherein said sub-windows include a "waiting" sub-window, a "working" sub-window and a "completed" sub-window.

3. (Original) The method of Claim 2 wherein the step of displaying the status of the execution of the command includes displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

4. (Original) The method of Claim 3 wherein when the command begins to execute on a computer system, the

name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

5. (Original) The method of Claim 4 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

6. (Original) The method of Claim 5 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

7. (Original) The method of Claim 6 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

8. (Previously presented) The method of Claim 7 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

9. (Previously presented) The method of Claim 8 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

10. (Original) The method of Claim 9 wherein when the displayed name of a computer system is selected

further information about the status of the command executing on the computer system is displayed.

11. (Original) The method of Claim 10 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

12. (Previously presented) The method of Claim 11 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

13. (Previously presented) A computer program product on a computer readable medium for displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said computer program product comprising:

code for displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

code for displaying the status of the execution of the command on each of the computer systems within the proper sub-window.

14. (Original) The computer program product of Claim 13 wherein said sub-windows include a "waiting" sub-

window, a "working" sub-window and a "completed" sub-window.

15. (Original) The computer program product of Claim 14 wherein the code for displaying the status of the execution of the command includes code for displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

16. (Original) The computer program product of Claim 15 wherein when the command begins to execute on a computer system, the name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

17. (Original) The computer program product of Claim 16 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

18. (Original) The computer program product of Claim 17 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

19. (Original) The computer program product of Claim 18 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

20. (Previously presented) The computer program product of Claim 19 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

21. (Previously presented) The computer program product of Claim 20 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

22. (Original) The computer program product of Claim 21 wherein when the displayed name of a computer system is selected further information about the status of the command executing on the computer system is displayed.

23. (Original) The computer program product of Claim 22 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

24. (Previously presented) The computer program product of Claim 23 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

25. (Previously presented) An apparatus for displaying an execution status of a command, said command being sent to a plurality of computer systems on a network for execution, said apparatus comprising:

means for displaying a dialog window, said dialog window being divided into sub-windows for displaying present status of the execution of the command on each of the computer systems; and

means for displaying the status of the execution of the command on each of the computer systems within the proper sub-window.

26. (Original) The apparatus of Claim 25 wherein said sub-windows include a "waiting" sub-window, a "working" sub-window and a "completed" sub-window.

27. (Original) The apparatus of Claim 26 wherein the means for displaying the status of the execution of the command includes means for displaying the names of the computer systems in the sub-windows in accordance with the status of the execution of the command on the computer systems.

28. (Original) The apparatus of Claim 27 wherein when the command begins to execute on a computer system, the name of the computer system is moved from the "waiting" sub-window to the "working" sub-window.

29. (Original) The apparatus of Claim 28 wherein when the command has finished executing on a computer, the name of the computer is moved from the "working" sub-window to the "completed" sub-window.

30. (Original) The apparatus of Claim 29 wherein the "completed" sub-window is further divided into a "successful" sub-window and a "failed" sub-window.

31. (Original) The apparatus of Claim 30 wherein the names of the computer systems that have successfully completed the execution of the command are displayed in the "successful" sub-window.

32. (Previously presented) The apparatus of Claim 31 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in the "failed" sub-window.

33. (Previously presented) The apparatus of Claim 32 wherein the names of the computer systems that have not successfully completed the execution of the command are displayed in red in the "failed" sub-window.

34. (Original) The apparatus of Claim 33 wherein when the displayed name of a computer system is selected further information about the status of the command executing on the computer system is displayed.

35. (Original) The apparatus of Claim 34 wherein if the selected computer system is displayed in the failed sub-window, a reason for the unsuccessful completion of the execution of the command is displayed.

36. (Previously presented) The apparatus of Claim 35 wherein if the selected computer system is displayed in the executing sub-window, a real-time progress of the execution of the command is displayed.

37. (Previously presented) A method of displaying an execution status of a command, the command being executed by a plurality of computer systems on a network, the computer systems running different system management software utilities having different command structures, the method comprising the steps of:

enabling a user to enter the command in a common interface, the command being either a request to start execution of another command or to stop execution of the other command, the common interface translating the command into the different command structures;

enabling a user to send the command to the plurality of the computer systems;

enabling a user to indicate whether or not the execution of the command is to be monitored;

displaying, if the execution of the command is to be monitored, a dialog window that is divided into a waiting, working, successful and failed sub-windows for displaying present status of the execution of the command on each of the computer systems executing the command; and

displaying the status of the execution of the command on each of the computer systems within a proper sub-window.

(IX)

Evidence Appendix

None.

(x)

Related Proceedings Appendix

None.